A parallel three-dimensional radial basis function interpolation method for unstructured dynamic meshes

O. Estruch*, O. Lehmkuhl*,**, R. Borrell** and C.D. Pérez Segarra*
Corresponding author: cttc@cttc.upc.edu

* Centre Tecnològic de Transferència de Calor (CTTC), Universitat Politècnica de Catalunya, Spain.
** Termo Fluids S.L..

Abstract: A radial basis function (RBF) interpolation method is implemented to be applied in computational fluid dynamics (CFD) problems with dynamic meshes. It is compared with the classic springs analogy smoothing, focusing on the mesh quality preserving and the computational cost. Moreover, the parallelization of the algorithms is critically analyzed. The main goal is to achieve simulations with meshes of millions of control volumes and run them with hundreds of processors, as well as to couple the adaptive of the grid to a CFD algorithm and extrapolate to a 3D application.

Keywords: Parallelization, Computational Fluid Dynamics, Mesh deformation, Moving grid, Radial Basis Function interpolation.

1 Introduction

In a great amount of engineering applications it is necessary to resort to dynamically updated meshes. Many of these are moving boundary problems, bio-fluid mechanics problems (*e.g.* blood flow through veins and arteries) and fluid-structure interaction (*e.g.* flutter simulation of wings, aerolastic stability of bridges or tall buildings, dynamic parachute-air interaction). To achieve a good performance of the computations, a robust, accurate and fast method is needed to redistribute the mesh according to the domain movement. In that sense, moving mesh approaches are a better choice than regenerating the grid each time. Therefore, the present study is focused on the moving grid techniques to be applied in the field of the computational fluid dynamics (CFD).

The main complexity of the moving grid approach is to find an optimum technique being suitable for all type of meshes and physical situations. At the same time, it should preserve as much as possible the quality of the mesh while keeping computational cost. Furthermore, a desired skill of the method is its easy implementation for parallel computation. Although all these aspects have been the purpose of most of the studies, it has not been designed a method with these properties altogether yet.

Basically, two strategies to deform mesh are used. The first exploits the connectivity of the internal grid points. Among this, the most popular is the spring analogy [1, 2, 3, 4, 5, 6, 7, 8], which is specially thought for unstructured meshes made of tetrahedrons. The main problem of these methods is that they involve large systems of equations, implying an important computational cost. Besides, with fine meshes or high amplitude movements, there can appear invalid meshes due to grid lines crossovers.

The second strategy is based on interpolation algorithms, such as transfinite interpolations [9], which are an efficient solution for structured meshes, or the mesh deformation based on radial basis function (RBF) interpolation [10, 11]. The latter has been recently studied for fluid-structure interaction computations, in which an interpolation problem is solved to transfer the displacements known at the boundary to the interior of the grid. This way, RBF's interpolation appears to be a less expensive method, since only three small systems of linear equations need to be solved, just involving the grid points of the boundary instead of the whole CFD grid like in the spring analogy.

To the authors knowledge, the existing sources of the RBF's interpolation technique are restricted to specific cases and the reference to three-dimensional simulations is scarce. Moreover, the parallelization of the algorithm is not developed yet. Since it appears to be an efficient and robust alternative, this work is focused on the RBF's interpolation. In this paper, this technique will be implemented and compared with the well-known spring analogy, in form of the lineal [2, 3, 6, 7, 8], torsional [3, 4, 5, 7, 8], semi-torsional [6, 7] as well as the ball-vertex [8] spring analogies. The mesh and case dependence are evaluated, together with the resultant grid quality, the computational requirements and the parallel performance. Furthermore, the parallelization of the algorithm and the extrapolation to three-dimensional cases represent the biggest challenges of the present work.

2 Mathematical formulation and numerical method

In this section is given a brief illustration of the RBF interpolation method. Being the deformable domain $\Omega \subset \mathbb{R}^d$, with $d \geq 2$, and $V = \{x_k\}_{k \in \Upsilon}$ the set of vertices of the CFD grid covering Ω, where $\Upsilon = \{1, \cdots, N_v\}$ is the set of indexes, with $|\Upsilon| = N_v$ the total number of vertices. We define the set of boundary vertices $V_b = \{x_i\}_{i \in \Upsilon_b}$, where $\Upsilon_b \subset \Upsilon$ and $|\Upsilon_b| = N_{v_b}$. Given the known discrete displacements of the vertices at the boundary,

$$u_i = g_i \quad \forall i \in \Upsilon_b \tag{1}$$

the interpolation space consists of all functions of the form

$$s(\mathbf{x}) = \sum_{i \in \Upsilon_b} \gamma_i \phi(\| \mathbf{x} - \mathbf{x}_i \|) + h(\mathbf{x}) \tag{2}$$

where ϕ is a radial basis function, h is a first degree polynomial, the coefficients $\gamma_i \in \mathbb{R}$ and $\| \cdot \|$ denotes the Euclidean norm. Note that for each coordinate direction we need an interpolation function. Then, the respective coefficients $\gamma = (\gamma_1, \cdots, \gamma_{N_{v_b}})^T$ and the polynomial h are determined by imposing the interpolation conditions

$$s(\mathbf{x}_i) = g_i \quad \forall i \in \Upsilon_b \tag{3}$$

$$\sum_{i \in \Upsilon_b} \gamma_i q(\mathbf{x}_i) = 0 \tag{4}$$

for all polynomials q with degree less or equal than that of polynomial h (*i.e.* first degree). Leading to the following linear system of equations,

$$\begin{pmatrix} \mathbf{M} & \mathbf{P} \\ \mathbf{P}^T & \mathbf{0} \end{pmatrix} \begin{pmatrix} \gamma \\ \beta \end{pmatrix} = \begin{pmatrix} \mathbf{g} \\ \mathbf{0} \end{pmatrix} \tag{5}$$

where $\mathbf{M}$ is the interpolation matrix of dimension $N_{v_b} \times N_{v_b}$ containing the evaluation of the basis function $M_{ij} = \phi(\| \mathbf{x}_i - \mathbf{x}_j \|)$, $i, j \in \Upsilon_b$, and $\mathbf{P}$ is a $N_{v_b} \times 4$ matrix with row i given by $(1 \; x_i \; y_i \; z_i)$. The vector $\mathbf{g} = \{g_i\}_{i \in \Upsilon_b}$ contains the known boundary displacements and β is the vector of coefficients for the polynomial h. Defined as,

$$h(\mathbf{x}) = \beta_1 + \beta_2 x + \beta_3 y + \beta_4 z \quad \forall \mathbf{x} = (x, y, z)^T \tag{6}$$

Common radial functions are shown in Table 1. The Table 1a presents several RBF's with compact support, *i.e.* , such that

$$\phi(x) = \begin{cases} f(x) & x \in [0,1] \\ 0 & otherwise \end{cases} \tag{7}$$

The RBF's in Table 1b are of global support. Thus, the sparsity of the interpolation matrix depends on the RBF employed. In many applications it is appropiate to scale with a support radius r to control the compact support, so $\phi \equiv \phi(x/r)$.

No.	Name	$f(\xi)$
1	CP C^0	$(1-\xi)^2$
2	CP C^2	$(1-\xi)^4(4\xi+1)$
3	CP C^4	$(1-\xi)^6(\frac{35}{3}\xi^2+6\xi+1)$

(a)

No.	Name	$f(x)$
4	Thin plate spline (TPS)	$x^2\log(x)$
5	Multiquadric biharmonics (MQB)	$\sqrt{1+x^2}$
6	Gaussian (GS)	e^{-x^2}

(b)

Table 1: Radial basis functions with a) compact support b) global support. Source: [10].

3 Description of the case

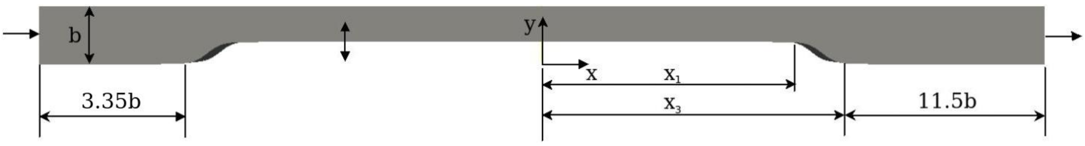


Figure 1: Geometry of the test case, with b=1.

With the purpose to analyze the moving grid methods itself, a benchmark problem based on a duct with a moving indentation has been selected [12, 13]. The geometric parameters and the moving law are shown in Figure 1 and Equation 8, respectively.

$$y(x) = \begin{cases} h & |x| \in [0,x_1] \\ 0.5h\left[1-tanh\left(a(|x|-x_2)\right)\right] & |x| \in [x_1,x_3] \\ 0 & |x| > x_3 \end{cases} \tag{8}$$

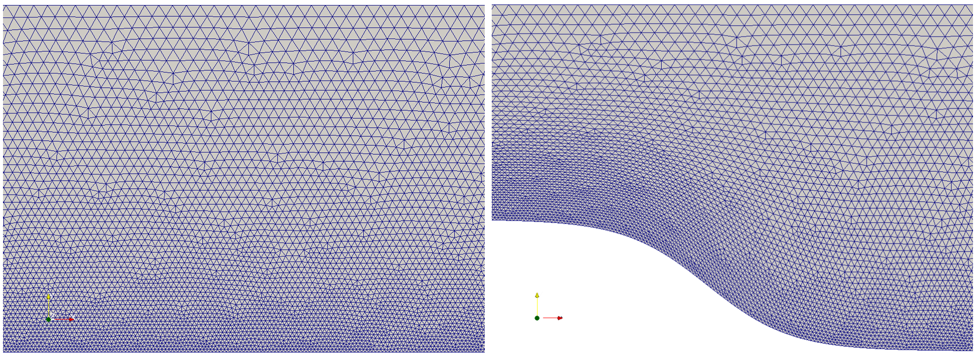
where $a = 4.14$, $x_1 = 4b$, $x_3 = 6.5b$, $x_2 = 0.5(x_1+x_3)$ and $h = 0.5h_{max}[1-cos(2\pi t^*)]$, with $h_{max} = 0.38b$ and $t^* = \frac{t}{T}$. The oscillation period T is obtained from the Strouhal number $St = \frac{b}{UT} = 0.037$.

4 Illustrative results and future lines

The numerical methodology applied in this work is based on finite volume technique. Governing partial differential equations are discretized using unstructured meshes by means of the three-dimensional CFD package Termo Fluids [14]. Second-order schemes are used for transient and spatial discretization. Furthermore, the mathematical formulation used in this work is based on symmetry preserving discretization of the governing equations adapted to moving boundary problems. Thus, some important properties of the Navier-Stokes equations are retained in the discretization process [15].

For the benchmark case, both the spring analogy and the RBF interpolation are studied using different meshes. In each test the quality of the deformed mesh is taken into account by means of the aspect ratio and, regarding the computational cost, it is quantified the CPU time invested in deforming the mesh. The performance is analyzed both in sequential and parallel modes, in order to extract the corresponding speed-ups and verify its efficient parallelization. All computations reported in this paper have been performed on a 76 nodes cluster, where each node has 2 AMD Opteron Quad-Core processors linked in an infiniband DPR4X network.

Relating to the quality of the mesh, the RBF interpolation obtains smoother meshes in comparison to the spring analogy. On the other hand, the spring analogy can result into very small aspect ratio meshes, which are invalid, when large deformations are required or fine meshes are used. However, this feature is dependent on the radius of the compact support of the RBF's and it is aimed to be analyzed in the future. In Figure 2 an example of the mesh deformation process by means of the RBF interpolation is depicted.



(a) (b)

Figure 2: Deformation of a mesh with the RBF interpolation method: a) $t^* = 0$, b) $t^* = 0.5$. A zoom of the duct is shown.

In Figure 3, the speed-ups obtained for an unstructured mesh of 72.300 control volumes for different number of processors are depicted. It must be emphasized that the speed-ups values are relative to the sequential performance. In that sense it is also attached a table with the corresponding CPU times (Table 2). Both sources allow to conclude that the RBF interpolation methodology is clearly much faster than the classic springs analogy. As the performance of the latter is worse in sequential, the speed-ups obtained are quite high, but in all cases the cost is far bigger than this required with the RBF method.

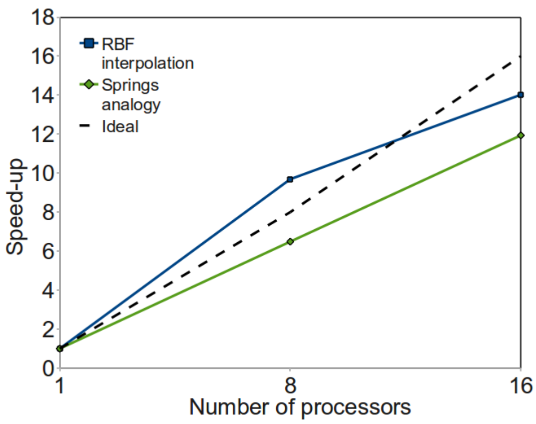


Figure 3: Parallelization of the methods.

Number of processors	RBF interpolation	Springs analogy	$\frac{t_{Springs}}{t_{RBF}}$
1	25.15	2780.49	110.56
8	2.60	428.53	165.03
16	1.79	232.93	129.81

Table 2: CPU times (s) invested in one iteration of the mesh deformation by each method.

The immediate goals to be carried out in the future are: improve the parallelization of the RBF method, bring the simulations to meshes of around millions of control volumes, run these with hundreds of processors, couple the adaptive of the grid to a CFD algorithm and extrapolate all this to a three-dimensional case.

5 Acknowledgements

This work has been financially supported by the Ministerio de Ciencia e Innovación of Spain (Contract No. ENE2010-17801) and Termo Fluids S.L. The authors thankfully acknowledge these institutions.

References

[1] C. FARHAT. Cfd on moving grids: from theory to realistic flutter, maneuvering, and multidisciplinary optimization. *International Journal of Computational Fluid Dynamics*, 19(8):595–603.

[2] F.J. BLOM. Considerations on the spring analogy. *International Journal For Numerical Methods in Fluids*, 32:647–668, 2000.

[3] CLARENCE O.E. BURG. A robust unstructured grid movement strategy using three-dimensional torsional springs.

[4] C. FARHAT, C. DEGAND, B.KOOBUS, and M. LESOINNE. Torsional springs for two-dimensional dynamic unstructured fluid meshes. *Computer Methods in Applied Mechanics and Engineering*, 163:231–245, 1998.

[5] C. DEGAND and C. FARHAT. A three-dimensional torsional spring analogy method for unstructured dynamic meshes. *Computers & Structured*, 80:305–316, 2002.

[6] M. MURAYAMA, K. NAKAHASHI, and K. MATSUSHIMA. Unstructured dynamic mesh for large movement and deformation. *AIAA Journal 2002-0122*, 2002.

[7] D. ZENG and C.R. ETHIER. A semi-torsional spring analogy model for updating unstructured meshes in 3d moving domains. *Finite Elements in Analysis and Design*, 41:1118–1139, 2005.

[8] C.L. BOTASSO, D. DETOMI, and R. SERRA. The ball-vertex method: a new simple spring analogy method for unstructured dynamic meshes. *Computer Methods in Applied Mechanics and Engineering*, 194(39-41):4244–4264, 2005.

[9] L.E. ERIKSSON. Generation of boundary-conforming grids around wing-body configurations using transfinite interpolation. *AIAA Jouranl*, 20(10):1313–2038, 1982.

[10] A. de BOER, M.S. van der SCHOOT, and H. BIJL. Mesh deformation based on radial basis function interpolation. *Computer & Structures*, 85:784–795, 2007.

[11] S. JAKOBSSON and O. AMOIGNON. Mesh deformation using radial basis functions for gradient-based aerodynamic shape optimization. *Computer & Fluids*, 36:1119–1136, 2007.

[12] I. DEMIRDZIĆ and M. PERIĆ. Finite volume method for prediction of fluid flow in arbitrarily shaped domains with moving boundaries. *International Journal for Numerical Methods in Fluids*, 10:771–790, 1990.

[13] P.J. ZWART, G.D. RAITHBY, and M.J. RAW. The integrated space-time finite volume method and its application to moving boundary problems. *Journal of computational physics*, 154:497–519, 1999.

[14] O. LEHMKUHL, C.D. PEREZ-SEGARRA, R. BORRELL, M. SORIA, and A. OLIVA. Termofluids: A new parallel unstructured cfd code for the simulation of turbulent industrial problems on low cost pc cluster. In *Proceedings of the Parallel CFD 2007 Conference*, pages 1–8, 2007.

[15] R.W. VERSTAPPEN and R.M. VAN DER VELDE. Symmetry-preserving discretization of heat transfer in a complex turbulent flow. *Journal of Engineering Mathematics*, 54(4):299–318, 2006.